AMENDMENTS TO THE CLAIMS

Please amend claims 14 and 26 as set forth below, without acquiescence in the Office Action's reasons for rejection or prejudice to pursue the original claims in a related application. A complete listing of the pending claims is provided below.

1. (Previously Presented) A method for enabling privileges comprising:

establishing a session on behalf of a user;

receiving a request to enable database privileges for the user;

upon receipt of the request to enable database privileges, verifying trusted security logic has previously been executed, wherein the act of verifying the trusted security logic comprises verifying a proxy user and checking a call stack; and

enabling database privileges for the user if the trusted security logic has previously been executed and is contained in one or more frames of the call stack.

2. (Previously Presented) The method of claim 1, further comprising:

storing call information in one or more frames of the call stack; and wherein

the act of verifying further comprises determining whether the one or more frames of the call stack corresponds to the trusted security logic.

3. (Previously presented) The method of claim 1, wherein the act of verifying the trusted security logic further comprises verifying an application name.

4. (Original) The method of claim 3, wherein the act of verifying the trusted security logic further includes verifying a security function name.

5. (Previously presented) The method of claim 1, wherein the act of verifying trusted security logic further comprises verifying a module name.

6. (Original) The method of claim 1, further comprising:

collecting one or more session parameters;

comparing the one or more session parameters against a set of trusted security parameters defined in a security function; and

returning a result indicating whether the one or more session parameters matches the set of trusted security parameters.

7. (Cancelled)

8. (Original) The method of claim 1, further comprising:
receiving information identifying the user;
prompting the user for a password;
authenticating the user based on information stored in an application program; and
associating the user with a role.

9. (Previously presented) A client-server computer system comprising:
a computer including:
a processor,
a main memory communicatively coupled to the processor; and
a disk storage communicatively coupled to the processor;

a database running on the computer from the main memory, the database further comprising:

one or more data structures stored in the disk storage, and
a call stack stored in the main memory;

an application program coupled to the database and configured to support a user; and

a metadata repository embodied in the one or more data structures stored in the disk storage, the metadata repository comprising trusted security logic; wherein

the application program is configured to initiate a call to enable database privileges, the call causing information to be stored in one or more frames of the call stack and one or more security functions to be executed; and wherein

the database is configured to:

verify the call stack comprises one or more frames corresponding to the trusted security logic;

test a proxy user; and

enable database privileges for the user if the trusted security logic is contained in the one or more frames of the call stack.

10. (Original) The client-server computer system of claim 9, wherein the application program resides with the database in the computer.

11. (Original) The client-server computer system of claim 9, wherein the application program resides on a separate computer communicatively coupled to the database.

12. (Original) The client-server computer system of claim 9, wherein the trusted security logic includes a schema name and a security package name.

13. (Cancelled)

14. (Currently Amended) A computer-readable medium that comprises a non-volatile medium or a volatile medium, having stored therein one or more sequences of instruction for enabling privileges, the one or more sequences of instructions causing one or more processors to perform a number of acts, said acts comprising:

establishing a session on behalf of a user;

receiving a request to enable database privileges for the user;

upon receipt of the request to enable database privileges, verifying trusted security logic has previously been executed, wherein the act of verifying the trusted security logic comprises verifying a proxy user and checking a call stack; and

enabling database privileges for the user if the trusted security logic has previously been executed and is contained in one or more frames of the call stack.

15. (Previously Presented) The computer-readable medium of claim 14, further comprising:

storing call information in one or more frames of the call stack; and wherein

the act of verifying further comprises determining whether the one or more frames of the call stack corresponds to the trusted security logic.

16. (Previously presented) The computer-readable medium of claim 14, wherein the act of verifying the trusted security logic further comprises verifying an application name.

17. (Original) The computer-readable medium of claim 16, wherein the act of verifying the trusted security logic further includes verifying a security function name.

18. (Previously presented) The computer-readable medium of claim 14, wherein the act of verifying trusted security logic further comprises verifying a module name.

19. (Original) The computer-readable medium of claim 14, further comprising:

collecting one or more session parameters;

comparing the one or more session parameters against a set of trusted security parameters defined in a security function; and

returning a result indicating whether the one or more session parameters matches the set of trusted security parameters.

20. (Cancelled)

21. (Original) The computer-readable medium of claim 14, further comprising:

receiving information identifying the user;

prompting the user for a password;

authenticating the user based on information stored in an application program; and

associating the user with a role.

22-25. (Cancelled)

26. (Currently Amended) A system for enabling privileges, wherein the system is implemented at least in part with a processor, the system comprising:

means for establishing a session on behalf of a user;

means for receiving a request to enable database privileges for the user;

means for upon receipt of the request to enable database privileges, verifying trusted security logic has previously been executed, wherein means for verifying the trusted security logic comprises means for verifying a proxy user and checking a call stack; and

means for enabling database privileges for the user if the trusted security logic has previously been executed and is contained in one or more frames of the call stack.

27. (Previously Presented) The system of claim 26, further comprising:

means for storing call information in one or more frames of the call stack; and wherein

means for verifying further comprises means for determining whether the one or more frames of the call stack corresponds to the trusted security logic.

28. (Previously presented) The system of claim 26, wherein means for verifying the trusted security logic further comprises means for verifying an application name.

29. (Previously presented) The system of claim 28, wherein means for verifying the trusted security logic further comprises means for verifying a security function name.

30. (Previously presented) The system of claim 22, wherein means for verifying trusted security logic further comprises means for verifying a module name.

31. (Previously presented) The system of claim 22, further comprising:

means for collecting one or more session parameters;

means for comparing the one or more session parameters against a set of trusted security parameters defined in a security function; and

means for returning a result indicating whether the one or more session parameters matches the set of trusted security parameters.

32. (Previously presented) The system of claim 22, further comprising:

means for receiving information identifying the user;

means for prompting the user for a password;

means for authenticating the user based on information stored in an application program; and

means for associating the user with a role.